Tutorial 8

Exercise 1

Given the following Python classes:

```
class Node:
    def __init__(self, el = None, n = None):
        self.next = n
        self.element = el

class LinkedStack:
   def __init__(self):
        self.top = None
        self.size = 0

   #return true if the stack is empty
   def empty(self):
     return self.size == 0

   #return the size of the stack
   def stack_size(self):
        return self.size

   #add the element to the top of the stack
   def push(self, element):
      #code to be completed in question (a)

   #return top element of stack
   def peek(self):
      #code to be completed in question (b)

   #remove top element of stack and return it
   def pop(self):
      #code to be completed in question (c)
```

```
class LinkedQueue:
   def __init__(self):
        self.front = None
        self.rear = None

   #return true if the queue is empty
   def isEmpty(self):
     return self.front == None

   #return the first element
   def getFrontElement(self):
     if (self.isEmpty()):
         return None;
     else:
         return self.front.element;

   #return the last element
   def getRearElement(self):
      #code to be completed in question (d)

   #add an element in the queue/enqueue
   def put(self, element):
      p = Node(element)
      if(self.isEmpty()):
        self.front = p   #empty queue
      else:
        self.rear.next = p    #nonempty queue
      self.rear = p

   #remove an element in the queue/dequeue
   def remove(self):
      #code to be completed in question (e)
```

Complete the following methods:

(a) The method `push(self, Object)` in `LinkedStack` class such that the method inserts the given object onto the top of the stack.

(b) The method `peek(self)` in `LinkedStack` class such that the method returns the object value on the top of the stack without removing it. When the stack is empty, `peek(self)` will return String object which stores "Empty Stack!" string.

(c) The method `pop(self)` in `LinkedStack` class such that the method returns the object on the top of the stack after removing it. When the stack is empty, `pop(self)` will return String object which stores "Empty Stack!" string.

(d) The method `getRearElement(self)` in the `LinkedQueue` class such that the method returns null if the queue is empty, otherwise returns the element at the rear of the queue.

(e) The method `remove(self)` method in the `LinkedQueue` class such that the method returns null if the queue is empty, otherwise removes an element from the front of the queue and returns the removed element.

Exercise 2

Complete method *reverseQ*, whose signature is given below. Method *reverseQ* should use a **Stack** to reverse the order of the items in its **Queue** parameter.

```
def reverseQ(q):
   //precondition: q contains x1 x2 ... xN  //(with x1 at the front)
   // postcondition: q contains xN ... x2 X1 //(with xN at the front)
```